Reference Energy Extremal Optimization: A Stochastic Search Algorithm Applied to Computational Protein Design

NAIGONG ZHANG, CHEN ZENG
Department of Physics, George Washington University, Washington, District of Columbia 20052

Received 19 February 2007; accepted in revised form 24 September 2007; accepted in final form 04 January 2008
DOI 10.1002/jcc.20937
Published online 19 March 2008 in Wiley InterScience (www.interscience.wiley.com).

Abstract: We adapt a combinatorial optimization algorithm, extremal optimization (EO), for the search problem in computational protein design. This algorithm takes advantage of the knowledge of local energy information and systematically improves on the residues that have high local energies. Power-law probability distributions are used to select the backbone sites to be improved on and the rotamer choices to be changed to. We compare this method with simulated annealing (SA) and motivate and present an improved method, which we call reference energy extremal optimization (REEO). REEO uses reference energies to convert a problem with a structured local-energy profile to one with more random profile, and extremal optimization proves to be extremely efficient for the latter problem. We show in detail the large improvement we have achieved using REEO as compared to simulated annealing and discuss a number of other heuristics we have attempted to date.

 J Comput Chem 29: 1762–1771, 2008

Key words: reference energy extremal optimization; extremal optimization; simulated annealing; combinatorial optimization; computational protein design

Introduction

Computational protein design seeks to use computational means to design amino acid sequences that can fold into a desired structure, or even, to achieve a desired function. In the past 10 years there have been significant achievements in this area, and a few landmark results are: the 1997 full-sequence redesign of a 28-residue zinc-finger structure without zinc,[1] the 2003 design of a 93-residue protein with a structure not previously seen in nature,[2] and the 2004 design of a biologically active enzyme (triose-phosphate isomerase).[3] For recent reviews, see Refs. 4–6.

In spite of the rapid progress and encouraging achievements, the dream of fully-automated, full-sequence design of an arbitrary structure still faces some significant challenges. One is the form of the energy function that can be used to select appropriately an amino acid sequence that is compatible with a given structure, and another is an efficient search algorithm that can then select an energetically optimal solution from an astronomically large number of protein sequence choices. In this article, we will focus on the second challenge, the computational search probelm in protein design. We introduce a new stochastic combinatorial optimization algorithm that, to our knowledge, is applied to computational protein design for the first time, and we show that, with proper consideration for the energy landscape of the problem, it performs significantly better than the search problem of choice, simulated annealing.

We will consider the search problem with a rigid backbone, with an energy function that can be written as single or pair-residue terms, and a discrete choice of amino acid conformations from a backbone-dependent rotamer library (i.e., a collection of energetically favorable and statistically significant side-chain conformations based on backbone torsion angles).[7] This goal is to find the global minimum energy conformation (GMEC), and this is the problem studied in Ref. 1 and in each fixed-backbone stage of Ref. 2. The search space here is enormous: with on average about 100 rotamers/backbone site, the design of a 80-residue protein will require searching through 100^{80} conformation choices. This search problem in fact has been shown to be NP-complete in Ref. 8, and Ref. 9 shows further that it is also hard to approximate with a theoretical guarantee. Despite the pronounced difficulties, a number of search algorithms have been used. Dead-end elimination (DEE) is the main method used in Ref. 1. It is a pruning algorithm that eliminates the rotamer choices that by energy comparison cannot be in the GMEC. In a series of papers,[10–13] the inequalities used for energy comparison have been progressively improved, resulting

This article contains supplementary material available via the internet at http://www.interscience.wiley.com/jpages/0192-8651/suppmat.

Correspondence to: C. Zeng; email: chenz@gwu.edu

Contract/grant sponsor: NSF; contract/grant numbers: DMR-0094175, DMR-0313129

in more and more rotamers eliminated. This method is deterministic and exact. Yet, with a fine choice of rotamers necessary for a realistic design problem, DEE often leaves too many states after all possible eliminations have been carried out, too many still for exhaustive search. Another class of search algorithms is stochastic, and the most efficient is simulated annealing (SA). A random walk is carried out in the conformation space using the Metropolis acceptance/rejection criterion with a gradual lowering temperature parameter. This is the method used in Ref. 2 (the code used in Ref. 2 has been released as the computer package RosettaDesign).[14,15]

Simulated annealing requires only the total energy of a particular conformation, and it either accepts or rejects this conformation based on this energy (and the current temperature). In cases where the energy function can be written as single and pair-residue terms, as is the problem studied here, the total energy can be written as a sum of local energies, and we have adapted a stochastic combinatorial optimization method that can use this local-energy information for efficient searches. In short, the method tries to improve on the backbone sites that have high local energies, and the new rotamers it selects are the ones with good local energies. Power-law probability distributions are used for site and rotamer selections, and this search algorithm, called extremal optimization (EO), has been shown to give comparable results with simulated annealing. This extremal optimization method was first developed for the hard optimization problems in the physics sub-field of spin glass by Boettcher and his co-workers.[16–20] It has also been applied to a number of classic optimization problems such as graph partitioning, graph coloring, and the traveling salesman problem with results comparable with and sometimes superior than those achieved with simulated annealing.[16] In this article, we adapt EO for protein design and study further the energy landscape of the problem. We realize that there is in fact an intrinsic structure in the local energies of the low-energy conformations; that is to say, the low-energy conformations have similar local-energy profiles that are consistent with the backbone shape and the characters of the physical interaction energies. This understanding aids our algorithm development significantly. By subtracting a set of reference local energies, we convert a structured local-energy profile to a random one, and the stochastic power of extremal optimization can then be fully used. We demonstrate that this new version of EO, which we call reference energy extremal optimization (REEO), is dramatically more efficient than the first adaptation of extremal optimization and simulated annealing. We believe that this result is a general one: for problems whose low-energy states have a structured local-energy profile, REEO will perform better than EO.

Methods

Test Proteins, Rotamer Choices, and Energy Function

We test our algorithms on a set of five proteins taken from Ref. 14. These are shown in Table 1 and are the ones whose sequences were successfully redesigned using the RosettaDesign program in Refs. 14, 15. As is customary in protein design calculations, a rotamer library is used to model the side-chain conformations of each amino acid, resulting in a discrete number of conformation choices. We use the standard Dunbrack backbone-dependent rotamer library[7] (release from May 2002) included in RosettaDesign. With all amino acids allowed at all positions, the default option in the program selects about 70 rotamers per residue (see Table 1).

Table 1. Five Test Proteins.

PDB code	Short name	Residue number (start-stop)	Rotamer number (N_{rot})
1HZ5	Protein L	62 (1–62)	4295
1AYE	Procarboxypeptidase	70 (10–79)	4812
1LMB	λ-repressor	87 (6–92)	5568
1URN	U1A	96 (2–97)	6475
2ACY	Acylphosphatase	98 (1–98)	6520

The energy function included in RosettaDesign has been described in detail in the supplementary materials of Refs. 2, 14. It is a linear sum of several terms including van der Waals, solvation, probablistic single-rotamer and pair-rotamer energies, and a reference energy for each of the 20 amino acids. In particular, the solvation energy uses the Lazaridis and Karplus implicit solvent model[23] which is a function of atom pairs. In fact, if we use i_r to denote the r-th rotamer at the backbone position i and j_s to denote the s-th rotamer at position j, all energy terms included in RosettaDesign are single-residue ($E(i_r)$) or pair-residue ($E(i_r,j_s)$) terms. For a particular rotamer choice, the total energy for this conformation is then

$$E = \sum_i E(i_r) + \sum_{i<j} E(i_r,j_s). \tag{1}$$

And if we define

$$\mathcal{E}(i_r,j_s) = E(i_r,j_s) + \frac{E(i_r)}{N-1} + \frac{E(j_s)}{N-1}, \tag{2}$$

where N is the total number of residues of our given backbone, then it is easy to show[24]

$$E = \sum_{i<j} \mathcal{E}(i_r,j_s), \tag{3}$$

which involves pairwise terms only.

With such a pairwise energy, we can convert our computational search problem for the GMEC to a graph problem.* In Figure 1, the large circles represent backbone positions (i and j), and the small circles represent rotamer choices (i_r and j_s). The weight on each edge is the energy $\mathcal{E}(i_r,j_s)$ [Eq. (2)]. There is no edge between two rotamers at the same site. Equation 3 then says that the total energy of a configuration is the total weight of a complete graph with one rotamer choice at each site. Our goal is therefore to find the minimum-weight complete graph among the astronomically large number of choices.

*Note this conversion to a graph problem can not be accomplished in a straightforward fashion for many-residue energy terms, for example a solvation energy that is the function of the total exposed (or buried) surface area. However, methods have been devised to convert this many-residue energy function to include single and pair-residue terms only. See ref. 21, 22.

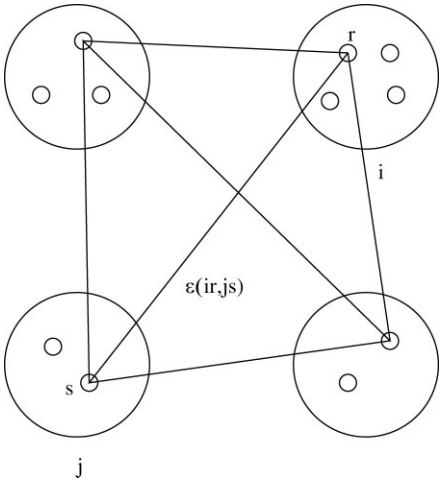

Figure 1. The search for GMEC can be considered as a search for the minimum weight complete graph. The large circles represent backbone sites (i and j), and the small circles represent rotamer choices (i_r and j_s). The weight on each edge is the energy $\mathcal{E}(i_r, j_s)$ [Eq. (2)]. There is no edge between two rotamers at the same site. Then Eq. (3) says that the total energy of the configuration is the total weight of a complete graph with one rotamer choice at each site.

For comparison a random-energy case RANDOM62 has also been studied. In this problem, the backbone and rotamer choices of 1HZ5 are used but the energies are random, *i.e.*, the energy $\mathcal{E}(i_r, j_s)$ on each graph edge [see Fig. 1 and Eq. (3)] is chosen randomly (and uniformly) between -1 and 1.

Simulated Annealing with Quenching

The search method of choice in RosettaDesign is simulated annealing (SA), which in fact does not require the energy to be pairwise (only the total energy of a conformation is needed). Starting from a random initial state, a random rotamer at a random site is selected for a new state. The energy of this new state is then calculated and compared with the energy of the present state, and the Metropolis criterion is used to decide whether the new state will be accepted or rejected. In simulated annealing, the temperature parameter in the Metropolis criterion is gradually reduced. At a higher temperatue, states with larger energy increases from the previous states can be accepted than at a lower temperature. This creates larger energy fluctuations in an attempt to escape from meta-stable energy wells in the search for the GMEC. As the temperature drops, the fluctuations become smaller as only smaller energy increases are allowed and the system settles into an energy well. Figure 2 shows the energy progression of a typical SA run.

In our program, the temperature is reduced from 100 to 0.3 kcal/mol, the same as in the RosettaDesign program, and it is reduced in a geometric fashion, with a specified reduction ratio. We have tested three temperature reduction ratios (1) 0.79 which results in 25 temperature cycles, $N_{\text{cycle}} = 25$; (2) 0.89 with $N_{\text{cycle}} = 50$; and (3) 0.943 with $N_{\text{cycle}} = 100$. To compare the three temperature reduction schedules, we fixed the total number of moves in each simulation to $2500 \times N_{\text{rot}}$ where N_{rot} is the total number of rotamers used for the protein to be designed. This means $100 \times N_{\text{rot}}$ moves per cycle for $N_{\text{cycle}} = 25$, $50 \times N_{\text{rot}}$ for $N_{\text{cycle}} = 50$, and $25 \times N_{\text{rot}}$ for $N_{\text{cycle}} = 100$.

At the end of each SA run, we also "quench" the final state; that is to say: (1) a random ordering of the backbone sites is chosen; (2) at each site all possible rotamers are selected one by one, with the rotamers at other residues fixed, and total energies are calculated; (3) the rotamer that produces the lowest total energy is chosen; (4) we continue this process until no lower energy is found for all sites of an ordering. This process ensures that no single-rotamer moves can lower the energy. It has been noted[25] that quenching can produce large improvement in the lowest energy found with modest additional cost in computer power. In our program, we have gone one step further by including a round of pair-rotamer quenching after exhaustive single-rotamer quenches: for each randomly chosen pair of backbone sites, all possible rotamer pairs are selected and the pair that gives the lowest energy (with rotamers at other sites fixed) is kept. Because pair quenching is expensive, we only carry out one round in which each site is quenched with each other site once in a random order.

The Extremal Optimization Method

In a series of papers, Boettcher and co-workers introduced a combinatorial optimization method called extremal optimization (EO),[16–20] originally applied to a class of hard optimization problems in a condensed matter physics sub-field called spin glass. In those problems there are usually a large number of positions, each position often has two state choices (often called up and down spins), and for each two positions is an interaction energy that depends on the spins at those positions. The goal is to find a low-energy conformation among an enormous number of configuration choices. Simulated annealing is also a well-tried method there, needing only the total energy of a conformation. The EO idea is to use the local energy information of the problem and improve on the sites that have high local energies.

1. Start with a random state.
2. Write the total energy as a sum of local energy terms $E = \sum_{i=1,\ldots,N} \mathcal{E}_i$.
3. Rank local energy $\mathcal{E}_i : \mathcal{E}_{\Pi(1)} \geq \mathcal{E}_{\Pi(2)} \geq \ldots \geq \mathcal{E}_{\Pi(N)}$ where k is the rank of site $\Pi(k)$.
4. Make a random change at site $\Pi(k)$ where the rank k is chosen from a power-law distribution with exponent $-\tau$: $P(k) \sim k^{-\tau}$, accept the change unconditionally, compute the new local energies, and go back to 3 for the next iteration.

The power-law distribution used in EO is biased toward the site with the highest local energy $\mathcal{E}_{\Pi(1)}$. It does not always improve on that site, as that will often result in the same site picked for change. Instead it gives all sites chances for change with a biase toward the high local-energy sites. It is found[16–20] that a τ roughly in a range from 1.1 to 1.6 is the best for many of the optimization problems studied, including graph partitioning, graph coloring, and spin glass (max-cut) problems. Note that $\tau = 0$ corresponds to choosing the position randomly (from a uniform distribution) and $\tau = \infty$ means choosing the highest local-energy position always. With this power-law distribution, Refs. 16–20 found that EO is often comparable with SA, and in some cases it performs better than SA.

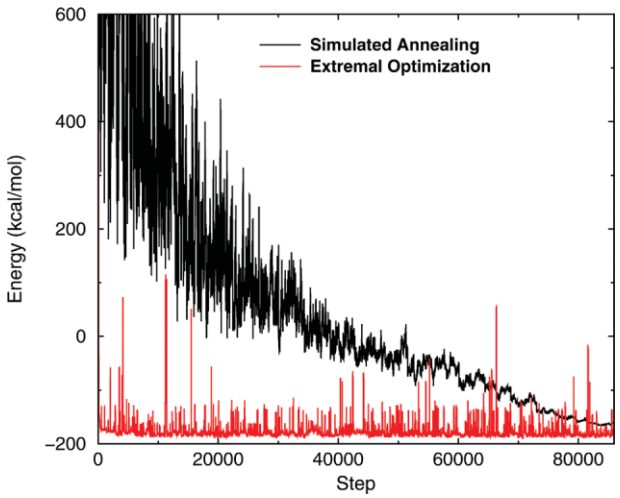

Figure 2. Energy progression for a typical simulated annealing and a typical extremal optimization run for a protein design problem. As temperature drops in a simulated annealing run, energy drops slowly, and fluctuations reduce. For an extremal optimization run, energy drops quickly in the initial stage, and large fluctuations are maintained throughout the simulation.

Our First Adaptation of Extremal Optimization for Protein Design

The adaptation of EO to computational protein design is straightforward. Our pairwise energy, Eq. (3), can be easily written in the local-energy form required by EO, if we define the local energy (often also called fitness, borrowing a term from evolution)

$$\mathcal{E}_i = \sum_{j \neq i} \mathcal{E}_{i,j}. \tag{4}$$

Strictly speaking $\sum_i \mathcal{E}_i$ from Eq. (4) equals two times the total energy in Eq. (3), but this will not affect the rankings in EO moves. On our graphical representation of energy in Figure 1, this local energy function $\mathcal{E}_i$ [Eq. (4)] is simply the sum of all weights on the "fan" of edges emanating from a rotamer on a particular site.

One complication from the protein design problem is that at each backbone site, we have many rotamer choices. In spin glass problem usually there are only two choices, and a move involves changing from the up spin to the down spin and vice versa. Here which rotamer should we choose to change into? A random choice of rotamers is an option. A choice of the rotamer that gives the best local energy is another. Borrowing from the idea of the EO for selecting sites, we can allow both to happen when we draw from a power-law distribution $(k')^{-\tau'}$ where k' is the rank of local energy for the rotamers at this site and τ' is a second exponent. Our extremal optimization method adapted for computational protein design is then:

1. Start with a random state.
2. Write total energy as a sum of local energy terms

$$2E = \sum_{i=1,\ldots,N} \mathcal{E}_i, \quad \mathcal{E}_i = \sum_{j \neq i} \mathcal{E}_{ij} \tag{5}$$

3. Rank local energy by site

$$\mathcal{E}_{\Pi(1)} \geq \mathcal{E}_{\Pi(2)} \geq \cdots \geq \mathcal{E}_{\Pi(N)} \tag{6}$$

where k is the rank of site index $\Pi(k)$.

4. Pick a rank from a power-law distribution with exponent $-\tau$: $P(k) \sim k^{-\tau}$ which is biased toward the high local energy sites

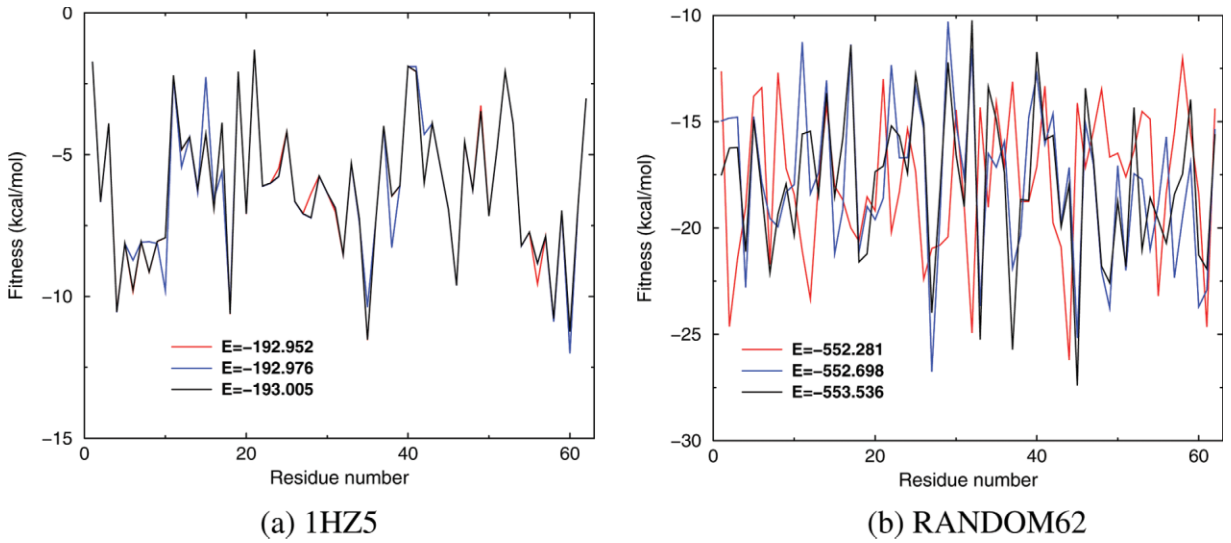

Figure 3. The local-energy profile of three low-energy states for (a) 1HZ5, using RosettaDesign energies, and (b) RANDOM62, with the same backbone and rotamer choices as 1HZ5 but using random energies. The realistic problem (a) shows a structured local-energy profile while the random-energy problem (b) shows a nonstructured local-energy profile.

5. At site $\Pi(k) = i$, rank the rotamer r using $\mathcal{E}_{i_r} = \sum_{j \neq i} \mathcal{E}_{i_r j_s}$ (with rotamers s at all other sites j fixed by the current state)

$$\mathcal{E}_{i_{\Pi'(1)}} \leq \mathcal{E}_{i_{\Pi'(2)}} \leq \cdots \leq \mathcal{E}_{i_{\Pi'(M_{\text{rot}}(i))}} \quad (7)$$

where k' is the rank of rotamer index $\Pi'(k')$ and $M_{\text{rot}}(i)$ is the number of rotamers at this site i.

6. Pick a rotamer index k' from a power-law distribution with exponent $-\tau'$: $P(k') \sim (k')^{-\tau'}$ which is biased toward the low local-energy rotamers.
7. Change the rotamer at site $\Pi(k)$ to rotamer $\Pi'(k')$, accept it unconditionally, and go back to 2 for the next iteration.

The two exponents, one for site selection (τ) and one for rotamer selection (τ'), need to be determined from optimization. Note that one step of an EO run will take more time than one step in a SA run. In a step of a SA run, only one total energy of the state under consideration is calculated. In a step of an EO run, local energies must be calculated and sorted and once a site is chosen for the move, local energies for all the rotamers at that site need to be calculated and then sorted.

We have used heap sort for both sorting steps in EO. In spin glass calculations, in which there are millions of sites and millions of local energies to be sorted, approximate heap sort has been found to be sufficient and efficient.[20] In our problem, our sorting need is divided into two parts: first sorting by site (the number of sites ranging from 62 to 98 for the five test proteins) and then sorting by rotamer (about 70 per site). We found that approximate heap sort does not save us much time and therefore have used exact heap sort for all EO calculations. Roughly, one EO move costs about the same time as 25 SA moves. For each run we have therefore used $100 \times N_{\text{rot}}$ EO moves as compared to $2500 \times N_{\text{rot}}$ SA moves.

Reference Energy Extremal Optimization

In spin glass problems, the interaction energies are random and all positions are essentially equal. In our protein design problems, on the other hand, the physical interaction energies (used in RosettaDesign) and the shapes of the protein backbone may result in certain sites having intrinsically lower (or higher) energies than others.

We looked into the low-energy states of 1HZ5 with realistic and random energies respectively. In Figure 3 we plot the local energies $\mathcal{E}_i$ for the three lowest total energies found in our simulations. It is clear that the local-energy profiles of the low-energy states in the realistic-energy problem (a) are structured, i.e., they are small perturbations of each other while maintaining an overall energy composition, with several well-defined valleys and peaks. On the other hand, the low-energy states in the random-energy problem do not exhibit a distinctive local-energy profile.

This local-energy profile complicates the EO searches. Sites with lower local energies will attract most improvement attempts, yet there is not much room for improvement. On the other hand, sites with higher local energies may in fact have more room for improvement. With this in mind, we developed a new method which we call reference energy extremal optimization (REEO). During site selection, instead of comparing and sorting local energy $\mathcal{E}_i$ we compare and sort the difference between the local energy and a reference local energy $\mathcal{E}^R_i$, i.e.,

$$\left(\mathcal{E}_{\Pi(1)} - \mathcal{E}^R_{\Pi(1)}\right) \geq \left(\mathcal{E}_{\Pi(2)} - \mathcal{E}^R_{\Pi(2)}\right) \geq \cdots \geq \left(\mathcal{E}_{\Pi(N)} - \mathcal{E}^R_{\Pi(N)}\right). \quad (8)$$

In this way, we compare the *potential* for improvement for these local energies (or the improvability of local energies) and improve on the sites with a higher potential for improvement.

Our next question is then what reference energy we should use. If we knew the GMEC, then using its local energies as reference energies would be the most efficient. Positions with local energy above that reference local energy will be improved on, and if we consider an EO run as a dynamical system, the reference state tends to serve as a target that can attract other states to it through EO moves. Without the knowledge of GMEC, we need to find reference energies that can approximate the energy profile of the GMEC. A natural choice is using the local energies of the lowest-energy state achieved so far in the simulation. Among several reference choices tested (this will be discussed later), this has proven to be the best.

Search Parameter Optimization

We now need to have a measure to determine for SA which N_{cycle} to use and for EO and REEO the power-law exponents τ and τ'. For each search method with generic parameters $\{P\}$, we carry out 100 independent runs for the i-th protein in our protein set and the average energy is denoted $E^{(i)}_{\text{avg}}(\{P\})$. And if we use $E^{(i)}_{\text{min}}$ to denote the minimum energy ever found for the i-th protein, then we can define a quality measure $D(\{P\})$ for the parameter set $\{P\}$:

$$D(\{P\}) = \sum_i \frac{E^{(i)}_{\text{avg}}(\{P\}) - E^{(i)}_{\text{min}}}{N_i} \quad (9)$$

where N_i is the number of residues for the i-th protein and i sums over the test protein set. What this measure does is that for a particular parameter set $\{P\}$ and for a test protein set, find the deviation from the minimum energy (ever found) per residue. We will use this measure to find the optimal N_{cycle} for SA and $\tau - \tau'$ combination for EO and REEO: the smaller $D(\{P\})$ the better.

Results

Simulated Annealing

The results for SA runs with ($N_{\text{cycle}} = 25, 50$, and 100) are shown in the supplementary materials, as are results of quenching after SA runs. Quenching improves results significantly for all temperature reduction ratios, more for the runs with fewer cycles than those with more cycles. This is reasonable as, with a sufficiently large number of steps per cycle, runs with a slower temperature reduction rate tend to explore the energy landscape more thoroughly. On the other hand, we observe that after single and pair-rotamer quenching, runs with fewer temperature cycles ($N_{\text{cycle}} = 25$) achieve slightly better results than those with more cycles ($N_{\text{cycle}} = 50, 100$). Using our measure Eq. 9, $D(N_{\text{cycle}} = 25) = 0.0611$, $D(N_{\text{cycle}} = 50) = 0.0715$, and $D(N_{\text{cycle}} = 100) = 0.0705$ (see Table 3).

Table 2. SA, EO, and REEO Search Results.

PDB code	Method	Average	Standard dev	Minimum	Search time	Quench time	τ	τ'
1HZ5	SA	−192.448	0.544	−193.005	101	44		
	EO	−192.492	0.580	−193.005	119	45	0.4	4.0
	REEO	−192.915	0.233	−193.005	121	45	1.4	2.5
1AYE	SA	−211.696	0.474	−212.177	133	72		
	EO	−211.784	0.349	−212.154	156	71	0.4	4.0
	REEO	−211.918	0.226	−212.082	136	71	1.4	2.5
1LMB	SA	−254.671	0.581	−255.571	202	135		
	EO	−254.607	0.624	−255.245	209	132	0.4	4.0
	REEO	−255.399	0.168	−255.571	195	133	1.4	2.5
1URN	SA	−278.762	0.855	−280.009	267	221		
	EO	−279.220	0.761	−280.009	274	213	0.4	4.0
	REEO	−279.943	0.156	−280.171	286	208	1.4	2.5
2ACY	SA	−301.069	1.322	−302.943	297	225		
	EO	−299.693	1.183	−302.375	290	217	0.4	4.0
	REEO	−302.078	0.806	−302.963	279	219	1.4	2.5
RANDOM62	SA	−520.842	6.568	−536.590	100	44		
	EO	−539.874	3.884	−550.652	118	45	1.0	2.5
	REEO	−537.921	4.379	−550.781	121	45	0.6	3.5

Average, standard deviation, and minimum energies are for 100 independent runs, including quenching, and are in kcal/mol. Search and quench time are in sec; Each calculation is done on a 2.8 GHz Xeon processor. The energy matrix $\mathcal{E}(i_r,j_s)$ was calculated and read into memory beforehand; this time was not included.

In Table 2, we show the average, standard deviation and minimum energy results for 100 independent independent SA runs (including single and pair-rotamer quenching). The temperature reduction ratio used is 0.79 ($N_{\text{cycle}} = 25$ with $100 \times N_{\text{rot}}$ moves per cycle; full results in supplementary materials). Search time is the average time (in sec) per run, and quench time is the average time (in sec) per run for single plus pair quenching. In our computer studies, a Linux cluster of 2.8 GHz Xeon processors was used. Each program is run on a single processor; no parallel processing is used except for running (independent) programs with different random seeds simultaneously on different processors. The pairwise energy matrix $\mathcal{E}(i_r,j_s)$ for each protein in Table 1 was calculated beforehand and stored in a file, involving about 10 million nonzero entries for 1HZ5 and about 20 million for 2ACY (this calculation took several hours for each protein). These energy matrices were read into memory before search was conducted; the reading time was less than 60 s and is not included in the times reported in Table 2 or subsequent tables on search results.

Extremal Optimization

The energy progression of a typical extremal optimization run for a protein design problem is shown in Figure 2. As compared to a typical simulated annealing run for the same problem, we see the same behavior as observed for SA/EO comparisons in other optimization problems (for example, see ref. 17). The EO run starts with high energies but very quickly finds some relatively low energy state. It then maintains relatively large fluctuations as it searches for the GMEC. The large fluctuations are possible because all states in the EO run are accepted unconditionally, i.e., there is no rejection of states as there is in simulated annealing. On the other hand, the fluctuations are also controlled by the τ and τ' parameters, as the move is biased toward the site with the worst local energy and the rotamer to be selected is biased toward the one with the best local energy.

We have also studied the effect of quenching at the end of EO runs (results in supplementary materials). Here single-rotamer quenching does not improve the results much. This is expected because single-rotamer quenching is simply an extremal optimization run with random site selection ($\tau = 0$) and best rotamer selection ($\tau' = \infty$). On the other hand pair-rotamer quenching does improve the average energy results.

The EO results for the five-protein set is shown in Table 2. As explained before we have chosen $100 \times N_{\text{rot}}$ EO moves that take about the same time as $2500 \times N_{\text{rot}}$ SA moves. For the five proteins with realistic energy, 56 $\tau - \tau'$ combinations have been tried ($\tau = 0, 0.2, 0.4, \ldots, 1.2$ and $\tau' = 1.5, 2.0, \ldots, 5.0$). To find a combination

Table 3. SA, EO, and REEO Parameter Optimization Using the Quality Measure $D(\{P\})$ [Eq. (9)] (in kcal/mol) and the five proteins in Table 1.

Method	Parameters	$D(\{P\})$
SA	$N_{\text{cycle}} = 25$	0.0611
	$N_{\text{cycle}} = 50$	0.0715
	$N_{\text{cycle}} = 100$	0.0705
EO	$\tau = 0.4, \tau' = 4.0$	0.0674
	$\tau = 0.6, \tau' = 3.5$	0.0690
	$\tau = 0.8, \tau' = 4.0$	0.0691
REEO	$\tau = 1.4, \tau' = 2.5$	0.0194
	$\tau = 1.8, \tau' = 2.5$	0.0205
	$\tau = 1.8, \tau' = 2.0$	0.0211

For each method the top 3 parameter combinations are shown. The measure is not very sensitive with parameter change. EO achieves comparable results to SA whereas REEO performs much better than SA and EO.

of $\tau - \tau'$ for our test protein set, we use the measure $D(\{P\})$ defined in Eq. (9). The three combinations that give the lowest $D(\{P\})$ are $D(0.4, 4.0) = 0.0674$, $D(0.6, 3.5) = 0.0690$, and $D(0.8, 4.0) = 0.0691$. Note these numbers are similar to what are achieved using SA (see Section "Simulated annealing" and Table 3).

Here EO achieves results comparable to SA and often slightly better results than SA. The optimal EO parameters found are $\tau = 0.4$ and $\tau' = 4.0$, *i.e.*, a fairly flat power-law for site selection and a fairly steep power-law for rotamer selection. In fact this combination resembles a slightly relaxed quench. (We have mentioned before that quench is $\tau = 0$ and $\tau' = \infty$.) The result is not sensitive to the $\tau - \tau'$ variation (see supplementary materials). The average, standard deviation, and minimum energy found for this optimal $\tau - \tau'$ combination are reported in Table 2.

For the random-energy case (RANDOM62), 128 $\tau - \tau'$ combinations have been tried, with $\tau = 0, 0.2, \ldots, 3.0$ and $\tau' = 2.0, 2.5, \ldots, 5.0$. Here, EO achieves much better results than SA. Average energy and minimum energy found are both significantly lower, and the standard deviation of the energy distribution has been reduced as well. The best exponents here are $\tau = 1.0$ and $\tau' = 2.5$, i.e., much closer to the $\tau = 1.1 - 1.6$ choices found in many of the random-energy optimization problems.[16–20]

The fact that EO achieves much better results than SA for the random-energy case and only slightly better for the realistic-energy problems is another evidence of the existence of an intrinsic local-energy profile for real proteins [as shown Fig. 3a], i.e., local energies at certain sites tend to be always higher than those at other sites, EO site selection then tends to pick these sites for improvement more often than others. It is therefore understandable why our optimized EO site-selection power-law exponent τ is so small (0.4–0.6); this helps to give a more even chance for site selection.

Reference Energy Extremal Optimization

Our REEO results using this choice of reference energy are also presented in Table 2. The same number of EO moves are attempted as the regular EO ($100 \times N_{\text{rot}}$), and 70 $\tau - \tau'$ combinations have been tried, with $\tau = 0.4, 0.6, \ldots, 3.0$ and $\tau' = 1.5, 2.0, \ldots, 3.5$. As in the case of EO, to find a combination of $\tau - \tau'$ for our test protein set, we use the measure $D(\{P\})$ defined in Eq. (9). The three combinations that give the lowest $D(\{P\})$ for REEO are $D(1.4, 2.5) = 0.0194$, $D(1.8, 2.5) = 0.0205$, and $D(1.8, 2.0) = 0.0211$.

As compared to EO we observe significant improvement in the average and standard deviation of the lowest energies found, and in $D(\{P\})$ measure (see Table 3). Visually, the results are dramatic. In Figure 4, the distributions of energies found for 100 SA, EO, and REEO runs, using the best parameters found for these methods and the test-protein set ($N_{\text{cycle}} = 25$ for SA, $\tau = 0.4$ and $\tau' = 4.0$ for EO, and $\tau = 1.4$ and $\tau' = 2.5$ for REEO), are plotted together for the five-protein set. REEO achieves signficiant improvement as compared to regular EO and SA. In particular, for the case of 1HZ5 (Fig. 4a), the lowest energy ever found in all simulation runs is -193.00548 kcal/mol. This energy is found 29 times in 100 SA runs, 41 times in 100 EO runs, and a significantly higher 83 times in 100 REEO runs (see supplementary materials).

The REEO results in Table 2 also shows that with the subtraction of reference local energies, the optimal exponents for EO now are $\tau = 1.4$ and $\tau' = 2.5$, much more like the exponents reported for random-energy problems than those quench-like combinations found with the first EO. This clearly shows that the use of reference energies indeed has converted a structured problem into a more random, unstructured problem, for which the power of the extremal optimization algorithm can be fully utilized.

In Table 2 and Figure 4f we also show REEO results for the random-energy case RANDOM62. The same 128 $\tau - \tau'$ combinations have been tried, as with the EO runs for RANDOM62, with $\tau = 0, 0.2, \ldots, 3.0$, and $\tau' = 2.0, 2.5, \ldots, 5.0$. Here the results are not as good as the straightforward EO results, and the exponents are more quench-like ($\tau = 0.6, \tau' = 3.5$).

We have also studied the effect of quenching on REEO (results in supplementary materials). Before any quenching, REEO achieves better average and minimum energy than regular EO. Single-rotamer quenching does little to EO or REEO, and pair quenching improves the results quite noticeably.

Discussion and Conclusions

Summary

We have studied the combinatorial optimization problem of computational protein sequence design with a fixed backbone. With a choice of rotamer library and a pair-residue energy function, as it is used in RosettaDesign, we have a minimum-weight graph search problem. We have adapted the extremal optimization algorithm, originally developed in the field of computational spin glass physics, to our search problem in protein design. We systematically explore the optimal exponent combinations and compare our results with those obtained from simulated annealing (with quenching). We notice a significant boost in performance when a reference energy is subtracted during the EO backbone site comparison process and show that this REEO performs significantly better than both SA and regular EO. We note that this is the case because we have converted a search problem with a structured local energy composition (due to the particular 3D structure of the backbone and the intrinsic physical properties of rotamer interactions) to one that is more random, and thus we are able to use the full power of the extremal optimization algorithm.

Can We Improve Our Results Further?

We have tried several other variations of REEO; none achieved better results than the one described earlier. We mention these ideas here as they may help the reader developing better algorithms.

We have tried two other choices for reference energy. The first is what we call rotamer minimum energy. For each rotamer, we can find the minimum local energy this rotamer can give by finding the minimum pairwise energies it forms with all rotamers at other sites of the protein. This energy gives a measure of lowest local energy this rotamer can achieve and can be used as reference energy when this rotamer appears in a state in the simulation. Another choice is what we call site minimum energy, which, for a particular site, is the minimum of all the rotamer minimum energies at this site.

We have also asked whether it is better to use the absolute value of this difference for comparison, i.e.,

$$\left|\mathcal{E}_{\Pi(1)} - \mathcal{E}^R_{\Pi(1)}\right| \geq \left|\mathcal{E}_{\Pi(2)} - \mathcal{E}^R_{\Pi(2)}\right| \geq \cdots \geq \left|\mathcal{E}_{\Pi(N)} - \mathcal{E}^R_{\Pi(N)}\right|. \quad (10)$$

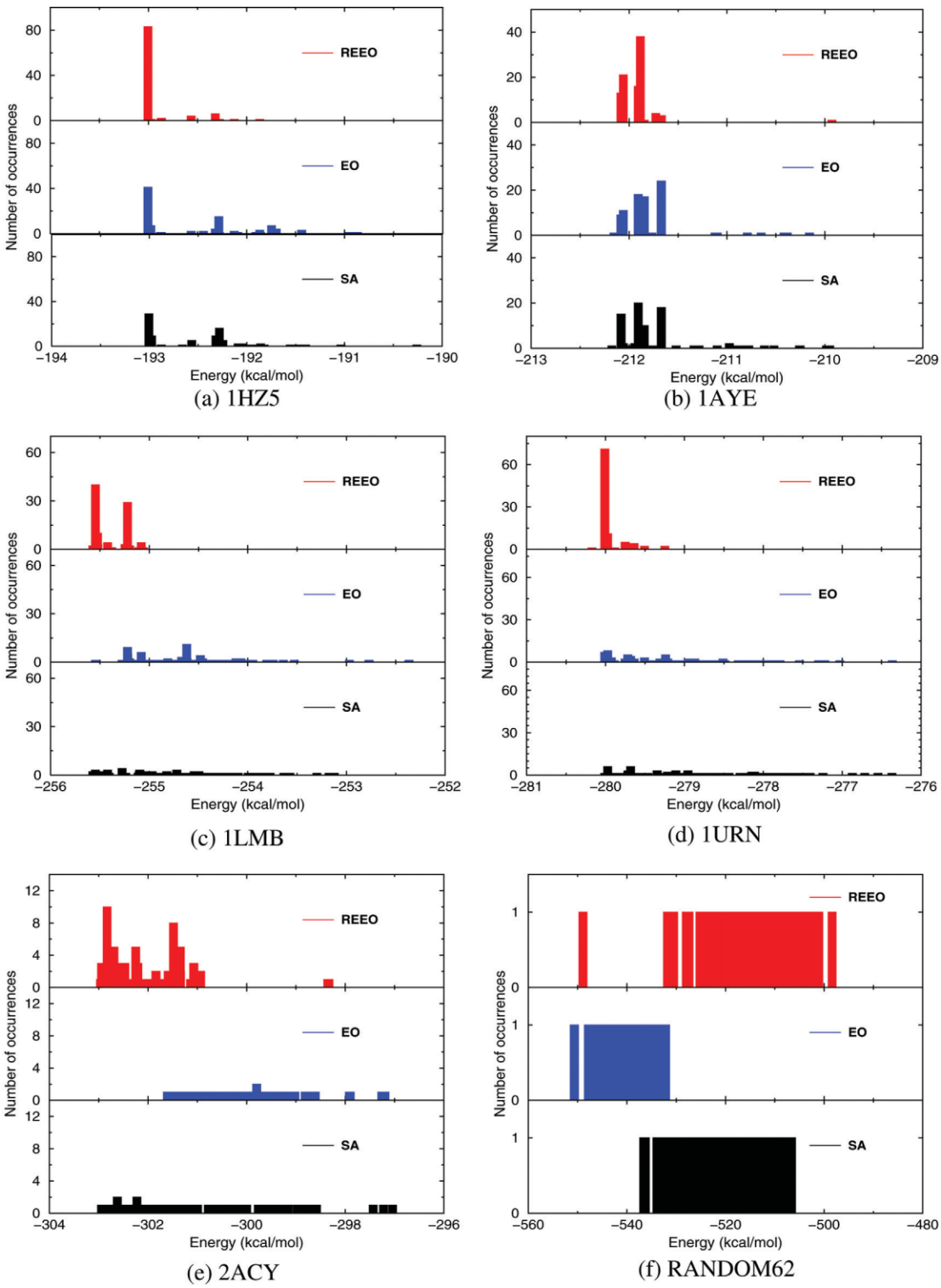

Figure 4. Energy distributions for 100 simulated annealing, extremal optimization, and reference energy extremal optimization runs, using $N_{\text{cycle}} = 25$ for SA runs and the best overall $\tau - \tau'$ combination for EO and REEO runs ($\tau = 0.4$ and $\tau' = 4.0$ for EO and $\tau = 1.4$ and $\tau' = 2.5$ for REEO, see Table 2).

This appears to have given more weight for changes to both the case when the local energy is much higher than the reference energy and the case when the local energy is much lower than the reference energy.

Our local energy is defined after scaling [Eq. (2)], through which process we eliminated the single-rotamer energies and placed their weights evenly on the edges linked to the rotamers. Because only the total energy is minimized in our problem, we also can imagine a different kind of scaling which absorbs more weight to the single-rotamer terms from pair-rotamer terms while keeping the total energy the same, trying, in fact, to make a many-body problem more like a one-body problem.

As shown in Figure 5, as we get to the later stages of an EO run, it is more and more difficult to find a state with a lower energy. The number of moves (shown in solid line) needed increases dramatically, and the number of sites that need to have rotamer changes from one lowest-energy state so far to another also increases. In early stages, it often take just tens of moves and just 2 or 3 site changes to produce a lower energy, but later it takes tens of thousands of moves and when a new lowest energy state is found about one-fifth of the sites have changed (13/62). In the physics literature,[17] these large-scale changes are called avalanches–a large cumulation of small moves that results in a large-scale change in the state of the system. Our question is then: is there a better way to identify more efficiently the coordinated moves needed to produce such an avalanche? Pair-rotamer quenching is in fact such a move, in which two rotamers change together. We have seen (in supplementary materials) that they produce rather significant gains. We have also explored the idea of decomposing the total energy in not local fan energies but energies of paths that can cover the complete graph. Sites along the path will then be able to change together. This idea is still at an early stage of development.

There is an interesting stochatic algorithm FASTER[26,27] that in fact makes a large number of moves at the same time. It is shown that with some improvement[27] this method performs significantly better

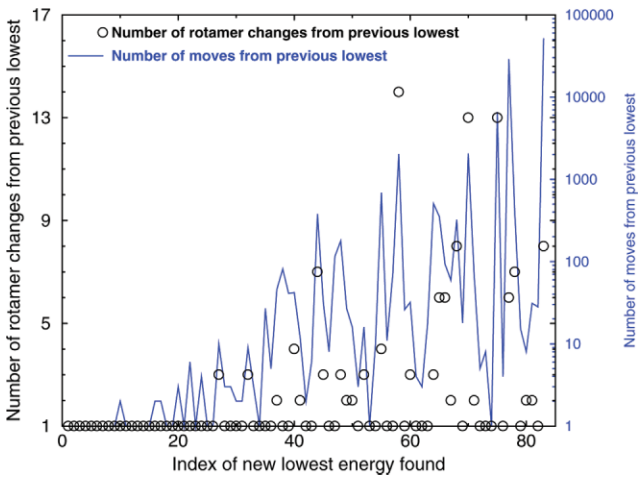

Figure 5. The number of rotamer changes (in circle) and the number of simulation moves taken from one current lowest energy found to the next are plotted for a typical extremal optimization run for the 62-residue 1HZ5 protein. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

than simulated annealing and can find low energy states extremely rapidly. We have done some preliminary study of this algorithm and hope to investigate its dynamics more in the future.

We have noted that quenching is fact a type of EO, with $\tau = 0$ and $\tau' = \infty$. In our EO and REEO, we start with a combination (τ, τ') and reach the quenching stage with $\tau = 0, \tau' = \infty$. We can imagine a gradually reduction of (τ, τ'), imitating the temperature reduction schedule in simuated annealing. We have not studied this implementation.

Finally, it is interesting to note that while here we use local energies extensively in EO and REEO searches, they have also been used to biase Monte Carlo and simulated annealing searches (see for example Refs. 28–30).

Conclusions

The hard optimization problems that EO has been applied to, such as spin glass, graph partitioning, graph coloring, and the traveling salesman problem, are all random-energy problems.[16] For these cases, EO has been shown to be comparable with or sometimes better than SA. For our design problem, with low-energy states having a consistent local-energy profile, the subtraction of reference local-energies, as carried out in REEO, achieves significantly better results than EO or SA. We believe our result is general. Further, the existence of such a low-energy local-energy profile (consistent with REEO results) suggests that a "funnel"-like energy landscape may exist for the fixed-backbone protein design problem, as has been suggested for the protein folding problem (see for example Ref. 31). That is to say, the overall funnel shape of the design/folding energy landscape (which is superimposed with smaller scale roughness of local maxima and minima) drives the dynamic process of design and folding, resulting in a stable design/folding sequence. Because of this dominant bias, the low-energy states are similar to the GMEC.

Acknowledgments

The authors thank Professor N.V. Fitton of Northern Virginia Community College for supplying us with the exact and approximate heap sort code and interesting discussions. They thank Ben Allen of Caltech for helpful discussions on the FASTER method. They also thank the referee for helpful comments and bringing to our attention Refs. 28–30.

References

1. Dahiyat, B. I.; Mayo, S. L. Science 1997, 278, 82.
2. Kuhlman, B.; Dantas, G.; Ireton, G. C.; Varani, G.; Stoddard, B. L.; Baker, D. Science 2003, 302, 1364.
3. Dwyer, M. A.; Looger, L. L.; Hellinga, H. W. Science 2004, 304, 1967.
4. Park, S.; Yang, X.; Saven, J. Curr Opin Struct Biol 2004, 14, 487.
5. Park, S.; Stowell, X.; Wang, W.; Yang, X.; Saven, J. Annu Rep Prog Chem Sect C 2004, 100, 195.
6. Butterfoss, G.; Kuhlman, B. Annu Rev Biophys Biomol Struct 2006, 35, 49.
7. Dunbrack, R. L.; Cohen, F. E. Prot Sci 1997, 6, 1661.
8. Pierce, N. A.; Winfree, E. Prot Eng 2002, 15, 779.
9. Chazelle, B.; Kingsford, C.; Singh, M. INFORMS J Comput 2004, 16, 380.

10. Desmet, J.; De Maeyer, M.; Hazes, B.; Lasters, I. Nature 1992, 356, 539.
11. Goldstein, R. F. Biophys J 1994, 66, 1335.
12. Pierce, N. A.; Spriet, J. A.; Desmet, J.; Mayo, S. L. J Comput Chem 2000, 21, 999.
13. Looger, L. L.; Helling, H. W. J Mol Biol 2001, 307, 429.
14. Dantas, G.; Kuhlman, B.; Callender, D.; Wong, M.; Baker, D. J Mol Biol 2003, 332, 449.
15. Liu, Y.; Kuhlman, B. Nucl Acids Res 2006, 34, W235.
16. Boettcher, S.; Percus, A. G. Artif Intellig 2000, 119, 275.
17. Boettcher, S. Comput Sci Eng 2000, 2, 75.
18. Boettcher, S.; Percus, A. G. Phys Rev Lett 2001, 86, 5211.
19. Boettcher, S.; Percus, A. G. Computational Modeling and Problem Solving in the Networked World; Kluwer, Boston, MA, 2003; pp. 61–77.
20. Boettcher, S. Eur Phys J B 2005, 46, 501.
21. Street, A. G.; Mayo, S. L. Fold Des 1998, 3, 253.
22. Zhang, N.; Zeng, C.; Wingreen, N. S. Proteins 2004, 57, 565.
23. Lazaridis, T.; Karplus, M. Proteins 1999, 35, 133.
24. Wernisch, L.; Hery, L.; Wodak, S. J. J Mol Biol 2000, 301, 713.
25. Voigt, C. A.; Gordon, D. B.; Mayo, S. L. J Mol Biol 2000, 299, 789.
26. Desmet, J.; Spriet, J.; Lasters, I. Proteins 2002, 48, 31.
27. Allen, B. D.; Mayo, S. L. J Comput Chem 2006, 27, 1071.
28. Cootes, A. P.; Curmi, P. M. G.; Torda, A. E. J Chem Phys 2000, 113, 2489.
29. Zou, J.; Saven, J. G. J Chem Phys 2003, 118, 3843.
30. Yang, X.; Saven, J. G. Chem Phys Lett 2005, 401, 205.
31. Onuchic, J. N.; Luthey-Schulten, Z.; Wolynes, P. G. Annu Rev Phys Chem 1997, 48, 545.